



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/653,618	08/31/2000	Cem Paya	MS	4621
23552	7590	01/11/2005	#147265.1/40062.67US01	
MERCHANT & GOULD PC P.O. BOX 2903 MINNEAPOLIS, MN 55402-0903			EXAMINER SHERKAT, AREZOO	
			ART UNIT 2131	PAPER NUMBER

DATE MAILED: 01/11/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/653,618

Applicant(s)

PAYA ET AL.

Examiner

Arezoo Sherkat

Art Unit

2131

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 July 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

Response to Amendment

This office action is responsive to Applicant's amendment received on July 9, 2004. Claims 1, 2, 7, 8, 13, 14, and 17-20 are pending.

The objection to claim 7 has been withdrawn due to Applicant's amendment.

Applicant's arguments filed on July 9, 2004 have been fully considered but they are not persuasive.

Response to Arguments

Applicant argues that Moudgill teaches away from inserting a security token into the stack as recited in claim 1, by pointing out the fact that a canary word may be guessed.

"Yet another approach is one in which the compiler puts a "canary" word just before the procedure return pointer on the stack. A canary word is simply a word containing a special pattern. Prior to returning from a routine, the code determines if the word has been overwritten. If so, it is determined that there has been a buffer overrun. Apart from requiring recompilation, this technique also suffers from the problem that it can be defeated by, e.g., guessing the canary word."

Examiner responds that Moudgill suggests that the technique of using a canary word, which is simply a word containing a **special pattern** suffers from the problem that it can be defeated by, e.g., guessing the canary word. Moudgill's disclosure lacks the required randomness of a security token, and he does acknowledge this problem. Nishikawa, on the other hand, teaches a semiconductor integrated circuit having a

Art Unit: 2131

diagnosis function including a scan chain arrangement block 2, in which a plurality of flip-flops are connected so that they can be shift-registered, and designed in a scan-path manner, a shift register 3 for storing required bits of a first random number pattern shifted by the block 2; another shift register 4 for storing required bits of a second random number pattern supplied to the block 2; and a comparator for comparing corresponding bits of the random number patterns stored in the shift registers 3 and 4 to detect whether all the bits of the random number patterns agree or disagree with each other ... (Nishikawa, abstract). The string of bits from shift register 3 can be pushed into the stack as a security token. This value may be popped from the stack into the shift register 4, and be compared against the content of the shift register 3 to detect buffer over flow.

Examiner respectfully maintains the rejection formulated on March 8, 2004.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moudgill, (U.S. Patent No. 6,578,094 and Moudgill hereinafter), in view of Nishikawa, (U.S. Patent No. 6,346,822 and Nishikawa hereinafter).

Regarding claim 1, Moudgill discloses a method for preventing overrun of an input data buffer within a program having the input data buffer on a stack data structure (i.e., stack allocated array/buffer), the program executing on a computing system, the method comprising:

pushing all arguments to a function onto the stack data structure, pushing a return address onto the stack data structure for use in obtaining the memory address for the instruction to be executed upon completion of the function (Col. 2, lines 10-47);

allocating memory locations on the stack data structure for use as local variables within the function (Col. 1, lines 1-40);

completing the instructions within the function (Col. 3, lines 3-10).

Moudgill further discloses preventing potentially overwriting a procedure return value due to array overflow by calling a "bounds checking procedure" that calculates and returns a safe upper bound value (Col. 5, lines 65-67 and Col. 6, lines 1-15).

Moudgill does not expressly disclose pushing onto the stack data structure a security token, the security token comprises a randomly generated data value, retrieving the security token value from the stack data structure, and if the retrieved security token value is identical to the randomly generated data value, return from the function using the return address stored on the stack data structure.

However, Nishikawa discloses a security token (i.e., a semiconductor integrated circuit), the security token comprises a randomly generated data value (i.e., a pseudo-random number pattern), retrieving the security token value from the stack data structure (i.e., shift register 3, where it is saved after being generated), and verifying if

the retrieved security token value is identical to the randomly generated data value (i.e., pseudo-random number pattern in comparison data register 11)(i.e., note that Fig. 1, element 2 is equivalent to the step of reading data into the input buffer in the way that it may influence the random number (pattern) to be changed from its original form), return from the function using the return address stored on the stack data structure (i.e., if it is verified that the pattern in shift register 3 is the same as the pattern in data register 11, it produces a success decision flag)(Col. 4, lines 38-67 and Col. 5, lines 1-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator, a storage unit to store the generated pattern/pseudo-random number, and at the same time store it on the stack, after allocated input data buffer, and then comparing the two numbers/patterns with each other to verify whether or not there has been an input buffer overflow with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 2, Moudgill discloses a method (i.e., bounds checking function) to prevent stack-smashing attacks.

Moudgill does not expressly disclose wherein the method further comprises aborting the operation of the program if the retrieved security token value is not identical to the randomly generated data value.

Art Unit: 2131

However, Nishikawa discloses wherein the method further comprises aborting the operation of the program if the retrieved security token value (i.e., random number pattern stored in the shift register) is not identical to the randomly generated data value (i.e., random number pattern stored in data register 11)(Col. 4, lines 65-67 and Col. 5, lines 1-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator, a storage unit to store the generated pattern/pseudo-random number, and at the same time store it on the stack, after allocated input data buffer, and then comparing the two numbers/patterns with each other to verify whether or not there has been an input buffer overflow with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 3, Moudgill does not expressly disclose wherein the randomly generated data value is determined using a random number generator once each time the program is executed.

However, Nishikawa discloses wherein the randomly generated data value is determined using a random number generator once each time the program is executed (i.e., after the reset signal is released)(Col. 3, lines 29-50).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator generating a random number pattern once each time the program is executed (i.e., after a reset signal is released) with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 5, Moudgill discloses wherein the function comprises a subroutine that does not return a data value (i.e., it is interpreted by the Office that the function that takes an array argument and calls the "gets (array argument)" routine may be designed to return or not to return a data value)(Col. 7, lines. 24-34)

Regarding claim 6, Moudgill discloses wherein the function comprises a subroutine that does returns one or more data values (i.e., "bounds ()" routine)(Col. 7, lines 55-67 and Col. 8, lines 1-49).

Regarding claim 7, Moudgill discloses an apparatus for preventing overrun of an input data buffer within a program having the input data buffer on a stack data structure (i.e., stack allocated array/buffer), the program, the apparatus comprising:

a function call module placing arguments to a function and a return address onto the stack data structure (Col.2, lines 10-47);

a perform function module performing the operations within the function, the perform function module allocates memory locations on the stack data structure for use as the input data buffer (Col. 1, lines 1-40);

a complete function module completing the operation of the function (Col. 3, lines 3-10).

Moudgill further discloses preventing potentially overwriting a procedure return value due to array overflow by calling a "bounds checking procedure" that calculates and returns a safe upper bound value (Col. 5, lines 65-67 and Col. 6, lines 1-15).

Moudgill does not expressly disclose pushing onto the stack data structure a security token, the security token comprises a randomly generated data value, retrieving the security token value from the stack data structure, and if the retrieved security token value is identical to the randomly generated data value, return from the function using the return address stored on the stack data structure.

However, Nishikawa discloses a security token (i.e., a semiconductor integrated circuit), the security token comprises a randomly generated data value (i.e., a pseudo-random number pattern), retrieving the security token value from the stack data structure (i.e., shift register 3, where it is saved after being generated), and verifying if the retrieved security token value is identical to the randomly generated data value (i.e., pseudo-random number pattern in comparison data register 11)(i.e., note that Fig. 1, element 2 is equivalent to the step of reading data into the input buffer in the way that it may influence the random number (pattern) to be changed from its original form), return from the function using the return address stored on the stack data structure (i.e., if it is

verified that the pattern in shift register 3 is the same as the pattern in data register 11, it produces a success decision flag)(Col. 4, lines 38-67 and Col. 5, lines 1-60).

The computer system of Moudgill may have been modified by Nishikawa to disclose:

a push security token module placing onto the stack data structure a security token, the security token comprises a randomly generated data value (i.e., a semiconductor integrated circuit, coupled to the computer system of Moudgill, comprising: a random number generator to generate a random number (pattern) and push it to the stack data structure of Moudgill's computer system), a pop security token module retrieving the security token from the stack data structure upon completion of the operation of the perform function module (i.e., retrieving the random number (pattern) from the shift register 3, where it is saved after being generated)(Nishikawa, Col.3, lines 29-63 and Col. 4, lines 38-65);

a test module comparing the retrieved security token with the randomly generated data value (i.e., verifying if the retrieved random number (pattern) is identical to the randomly generated data value in comparison data register 11- note that Fig. 1, element 2 is equivalent to the step of reading data into the input buffer in the way that it may influence the random number (pattern) to be changed from its original form), and wherein the complete function module returns from the function if the retrieved security token is determined to be identical to the randomly generated data value by the test module (i.e., if it is verified that the pattern in shift register 3 is the same as the pattern

in data register 11, it produces a success decision flag)(Nishikawa, Col. 3, lines 63-67 and Col. 4, lines 65-67 and Col. 5, lines 1-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator, a storage unit to store the generated pattern/pseudo-random number, and at the same time store it on the stack, after allocated input data buffer, and then comparing the two numbers/patterns with each other to verify whether or not there has been an input buffer overflow with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 8, Moudgill discloses a method (i.e., bounds checking function) to prevent stack-smashing attacks.

Moudgill does not expressly disclose wherein the complete function module aborts the operation of the program if the retrieved security token (i.e., random number pattern stored in the shift register) is determined not to be identical to the randomly generated data value (i.e., random number pattern stored in data register 11) by the test module (i.e., the comparator 5)(Col. 4, lines 65-67 and Col. 5, lines 1-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator, a storage unit to store the

generated pattern/pseudo-random number, and at the same time store it on the stack, after allocated input data buffer, and then comparing the two numbers/patterns with each other to verify whether or not there has been an input buffer overflow with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 9, Moudgill does not expressly disclose wherein the randomly generated data value is determined using a random number generator module once each time the program is executed.

However, Nishikawa discloses wherein the randomly generated data value is determined using a random number generator module once each time the program is executed (i.e., after the reset signal is released)(Col. 3, lines 29-50).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator generating a random number pattern once each time the program is executed (i.e., after a reset signal is released) with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 11, Moudgill discloses wherein the function comprises a subroutine that does not return a data value (i.e., it is interpreted by the Office that the function that takes an array argument and calls the "gets (array argument)" routine may be designed to return or not to return a data value)(Col. 7, lines. 24-34)

Regarding claim 12, Moudgill discloses wherein the function comprises a subroutine that does return one or more data values (i.e., "bounds ()" routine)(Col. 7, lines 55-67 and Col. 8, lines 1-49).

Regarding claim 13, Moudgill discloses a computer program product readable by a computing system and encoding a set of computer instructions for preventing overrun of an input data buffer within a program having the input data buffer on a stack data structure (i.e., stack allocated array/buffer), the program executing on a computing system, the method comprising:

pushing a return address onto the stack data structure for use in obtaining the memory address for the instruction to be executed upon completion of the function (Col. 2, lines 10-47);

completing the instructions within the function (Col. 3, lines 3-10).

Moudgill further discloses preventing potential overwriting of a procedure return value due to array overflow by calling a "bounds checking procedure" that calculates and returns a safe upper bound value (Col. 5, lines 65-67 and Col. 6, lines 1-15).

Moudgill does not expressly disclose pushing onto the stack data structure a security token, the security token comprises a randomly generated data value, retrieving the security token value from the stack data structure, and if the retrieved security token value is identical to the randomly generated data value, return from the function using the return address stored on the stack data structure.

However, Nishikawa discloses a security token (i.e., a semiconductor integrated circuit), the security token comprises a randomly generated data value (i.e., a pseudo-random number pattern), retrieving the security token value from the stack data structure (i.e., shift register 3, where it is saved after being generated), and verifying if the retrieved security token value is identical to the randomly generated data value (i.e., pseudo-random number pattern in comparison data register 11)(i.e., note that Fig. 1, element 2 is equivalent to the step of reading data into the input buffer in the way that it may influence the random number (pattern) to be changed from its original form), return from the function using the return address stored on the stack data structure (i.e., if it is verified that the pattern in shift register 3 is the same as the pattern in data register 11, it produces a success decision flag)(Col. 4, lines 38-67 and Col. 5, lines 1-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator, a storage unit to store the generated pattern/pseudo-random number, and at the same time store it on the stack, after allocated input data buffer, and then comparing the two numbers/patterns with each other to verify whether or not there has been an input buffer overflow with the

Art Unit: 2131

motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 14, Moudgill discloses a method (i.e., bounds checking function) to prevent stack smashing attacks.

Moudgill does not expressly disclose wherein the method further comprises aborting the operation of the program if the retrieved security token value is not identical to the randomly generated data value.

However, Nishikawa discloses wherein the method further comprises aborting the operation of the program if the retrieved security token value (i.e., random number pattern stored in the shift register) is not identical to the randomly generated data value (i.e., random number pattern stored in data register 11)(Col. 4, lines 65-67 and Col. 5, lines 1-60).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator, a storage unit to store the generated pattern/pseudo-random number, and at the same time store it on the stack, after allocated input data buffer, and then comparing the two numbers/patterns with each other to verify whether or not there has been an input buffer overflow with the motivation to provide a diagnosis function capable of diagnosing the operation state of a

semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 15, Moudgill does not expressly disclose wherein the randomly generated data value is determined using a random number generator once each time the program is executed.

However, Nishikawa discloses wherein the randomly generated data value is determined using a random number generator once each time the program is executed (i.e., after the reset signal is released)(Col. 3, lines 29-50).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator generating a random number pattern once each time the program is executed (i.e., after a reset signal is released) with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit (i.e., a perform function module)(Nishikawa, Col. 1, lines 40-45).

Regarding claims 4, 10, and 16, Moudgill does not expressly disclose wherein the random number generator generates the randomly generated data value using a snapshot of a system clock within the computing system before the program first accepts input data.

However, Nishikawa discloses wherein the random number generator (Fig. 1, element 1) generates the randomly generated data value using a snapshot of a system clock (i.e., clock counter, Fig. 1) within the computing system (Col. 3, lines 29-67 and Col. 4, lines 1-20).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill with the teachings of Nishikawa to include a pseudo-random number generator generating a random number pattern, initiated (i.e., after a reset signal is released) and ceased (i.e., when the signal of agreement of all the bits or the signal of disagreement of all the bits) by a clock counter before the program first accepts input data with the motivation to provide a diagnosis function capable of diagnosing the operation state of a semiconductor integrated circuit coupled to a computer system (i.e., verifying if the retrieved random number (pattern) is identical to the randomly generated data value in comparison data register 11- note that Fig. 1, element 2 is equivalent to the step of reading data into the input buffer in the way that it may influence the random number (pattern) to be changed from its original form)(Nishikawa, Col. 1, lines 40-45).

Regarding claim 17, Moudgill discloses wherein the function comprises a subroutine that does not return a data value (i.e., it is interpreted by the Office that the function that takes an array argument and calls the "gets (array argument)" routine may be designed to return or not to return a data value)(Col. 7, lines. 24-34)

Regarding claim 18, Moudgill discloses wherein the function comprises a subroutine that does return one or more data values (i.e., "bounds ()" routine)(Col. 7, lines 55-67 and Col. 8, lines 1-49).

Regarding claim 19, Moudgill discloses wherein the computer data product comprises a set of computer instructions encoded (i.e., programs written in programming languages such as C, C++, or Java) and stored onto a computer readable storage medium (i.e., memory)(Col. 1, lines 15-40).

Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Moudgill, (U.S. Patent No. 6,578,094 and Moudgill hereinafter) and Nishikawa, (U.S. Patent No. 6,346,822 and Nishikawa hereinafter), in view of Williams, (U.S. Patent No. 6,519,702 and Williams hereinafter).

The teachings of Moudgill and Nishikawa have been discussed previously.

Regarding claim 20, Moudgill or Nishikawa does not expressly disclose wherein the computer data product comprises a set of computer instructions encoded within a carrier wave for transmission between computing systems.

However, Williams discloses wherein the computer data product comprises a set of computer instructions encoded within a carrier wave for transmission between computing systems (Col. 2, lines 58-67).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of applicant's invention to modify the teachings of Moudgill and Nishikawa with the teachings of Williams to include the capability to communicate computer instruction signals in a carrier wave with the motivation to provide for the capability to execute not only the computer code stored on a computer readable storage medium, but also the computer code embedded in data received from an external source in an electronic form (Williams, Col. 1, lines 10-15).

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Arezoo Sherkat whose telephone number is (571) 272-3796. The examiner can normally be reached on 8:00-4:30 Monday-Friday.


Art Unit: 2131

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on (571) 272-3795. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Arezoo Sherkat
Patent Examiner
Group 2131
Jan 3, 2005


EMMANUELL L. MOISE
PRIMARY EXAMINER